

# **Deploying And Versioning Your Application**

**Steven Pratschner  
Pleas**

**Keith**

**Program Manager  
Pleas & .NET Framework  
Associates**

**Keith**

**1 Microsoft Corporation**

Microsoft®  
**PDC** 2000  
Professional Developers Conference

Microsoft®  
**.net**

the defining

***point***

# Agenda

- **The Problem**
- **The Solution**
- **Assemblies: The Building Block**
- **Versioning and Sharing**
- **Version Policy**
- **Deployment and Packaging Scenarios**
- **Lots of demos...**

# The Problem

## Versioning

- **Customer Perspective:**
  - Installation of one app can break others
  - Limited to one version of an app/component
- **Causes:**
  - Version dependencies are not tracked or enforced
  - Apps are not isolated
  - Side by side support not built in

# The Problem

## Deployment and Installation

- **Customer Perspective:**
  - Registry entries are complicated and error prone
  - Replicating apps is not straightforward
  - Deleting an app is not predictable
  - Updating an app is hard to do “on the fly”
- **Cause:**
  - Apps are not self-describing or self-contained



# **The Solution**

## **What's required**

- **Apps must be self describing**
- **Enable zero-impact install**
- **Version dependencies must be recorded and enforced**
- **Must remember “last known good”**
- **Support side by side components**
- **Application isolation**

# **Assemblies**

## **The Building Block**

- **Deployment unit for types and resources: a “logical dll”**
- **Self describing through a manifest**
- **Fundamental unit of versioning, reuse, deployment and permission grants and requests**
- **Core to type loading and execution**
  - **Visibility boundary**

# **Assemblies**

## **Assembly Manifests**

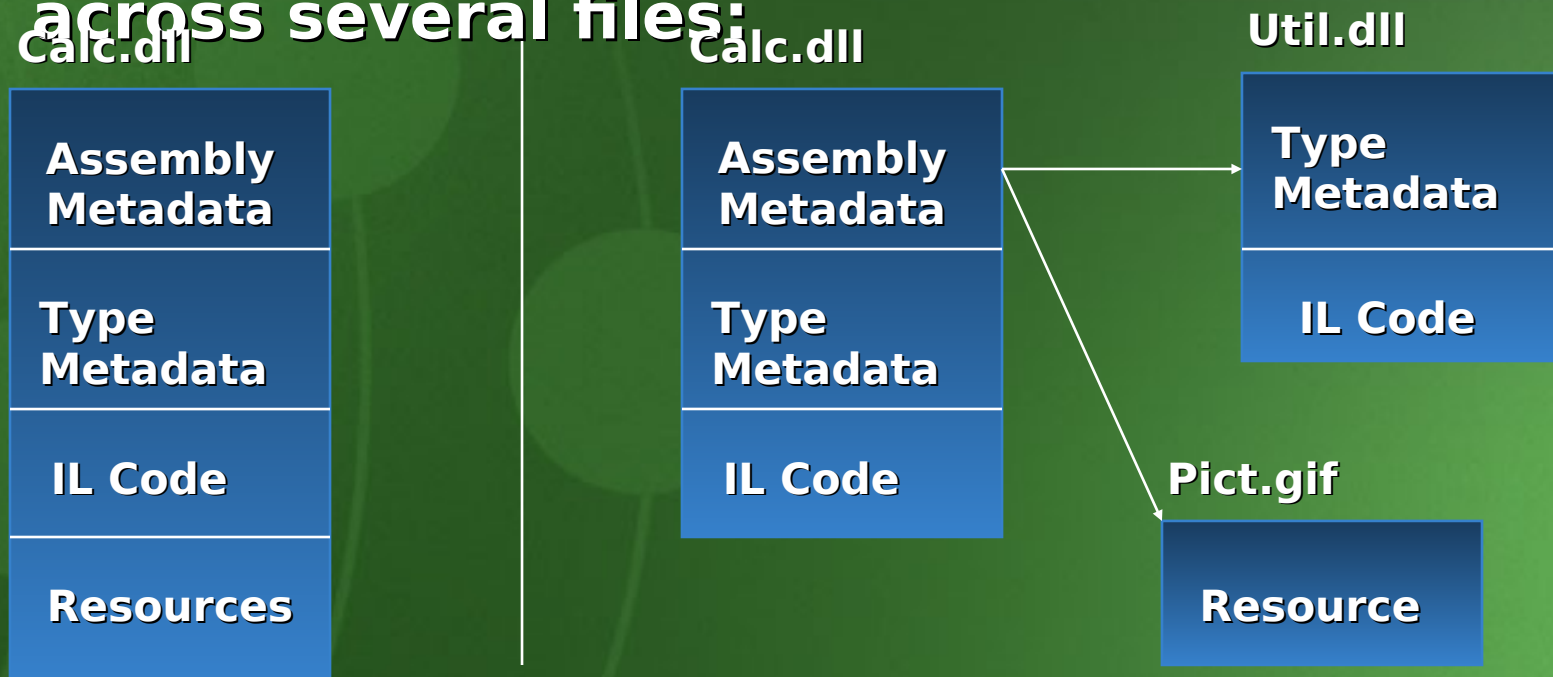
- **Metadata describing the assembly**
- **Manifest includes:**
  - **Identity: name + version + culture**
  - **File list**
  - **Referenced assemblies**
  - **Exported Types and Resources**
  - **Permission requests**



# Assemblies


## Assembly Structure

- Assemblies contain: Assembly Metadata (manifest), Type Metadata, IL code, Resources
- May all be in one file, or may be spread across several files:



# Assemblies

## Demo: Assembly Tools

-  **ILDASM.EXE**
  - Single assembly
  - c:\Program Files\NGWSSDK\Bin\Ildasm.exe
  - /adv switch for “advanced” options
- **APDEPENDS.EXE**
  - Assembly dependency hierarchies
  - PDC Conference CD, with 3-215 samples
  - Will ship with Beta 1 SDK

# Versioning And Sharing

- Isolate components when possible
  - Sharing is not the default
- Shared components are side by side
  - Allows choice of version
- Allow admin to change version policy

# **Versioning And Sharing Application- Private Assemblies**

- **Visible only to the containing application**
- **Simple naming requirements**
- **Deployed in the directory structure of the application**
  - **“Probed” at runtime**
- **No version checking done, no version policies apply**

# Versioning And Sharing

## Shared Assemblies

- Assembly that is shared among several applications on the machine
- More stringent naming requirements (Shared Names)
  - Uniqueness
  - Protection of the name
- Typically deployed to the global assembly store
  - May also be “privatized” to app directory
- Versions are checked, version policies apply



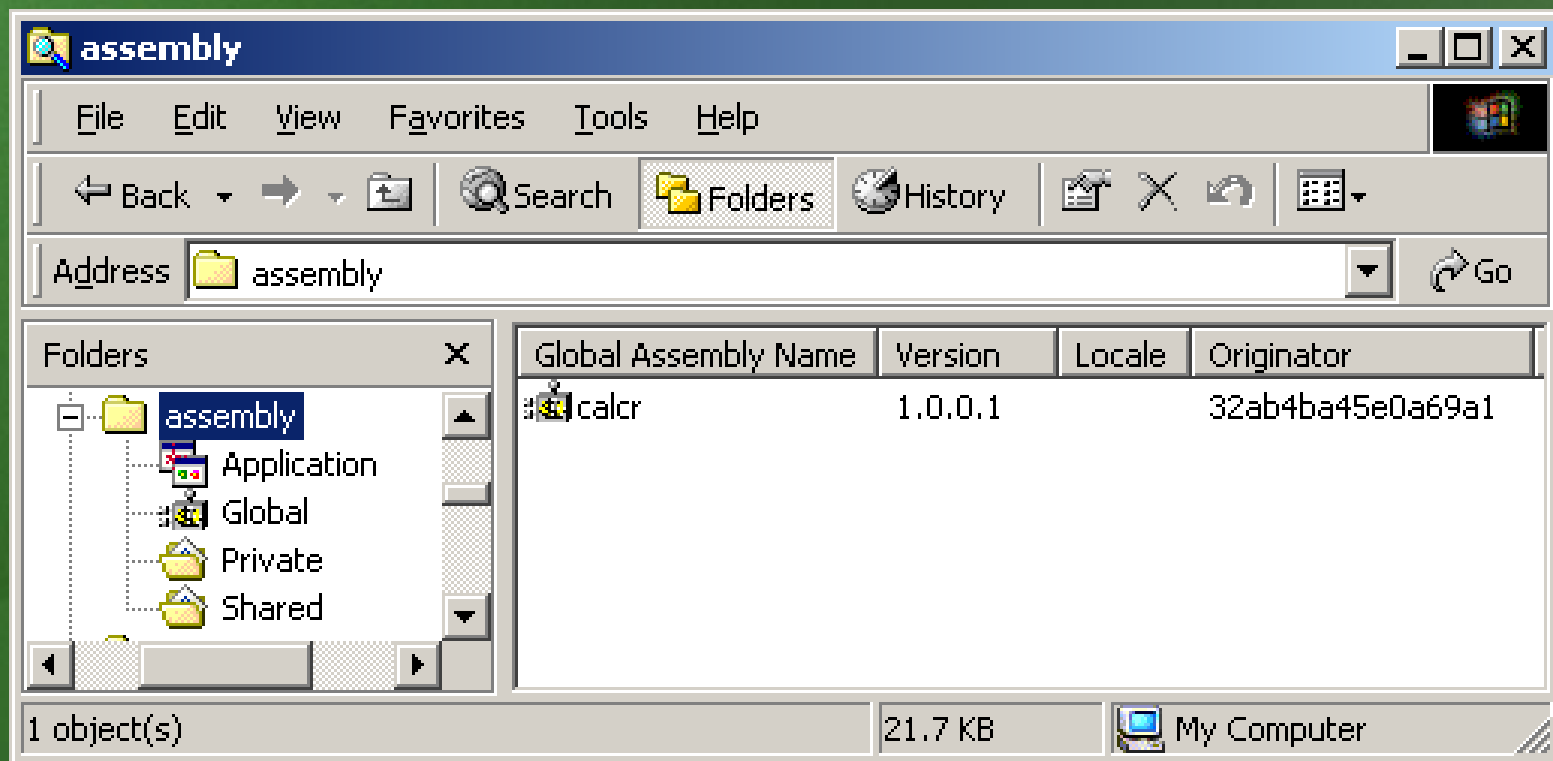
# Versioning And Sharing

## Global Assembly Store

- Machine-wide store for shared assemblies
- Machine-wide impact
- Assemblies in the store must have shared names
- Side by side storage
- Install/Uninstall by explicit action
  - Installers, including Windows Installer

# Versioning And Sharing

## Demo: Global Assembly Store Viewer



# Versioning And Sharing

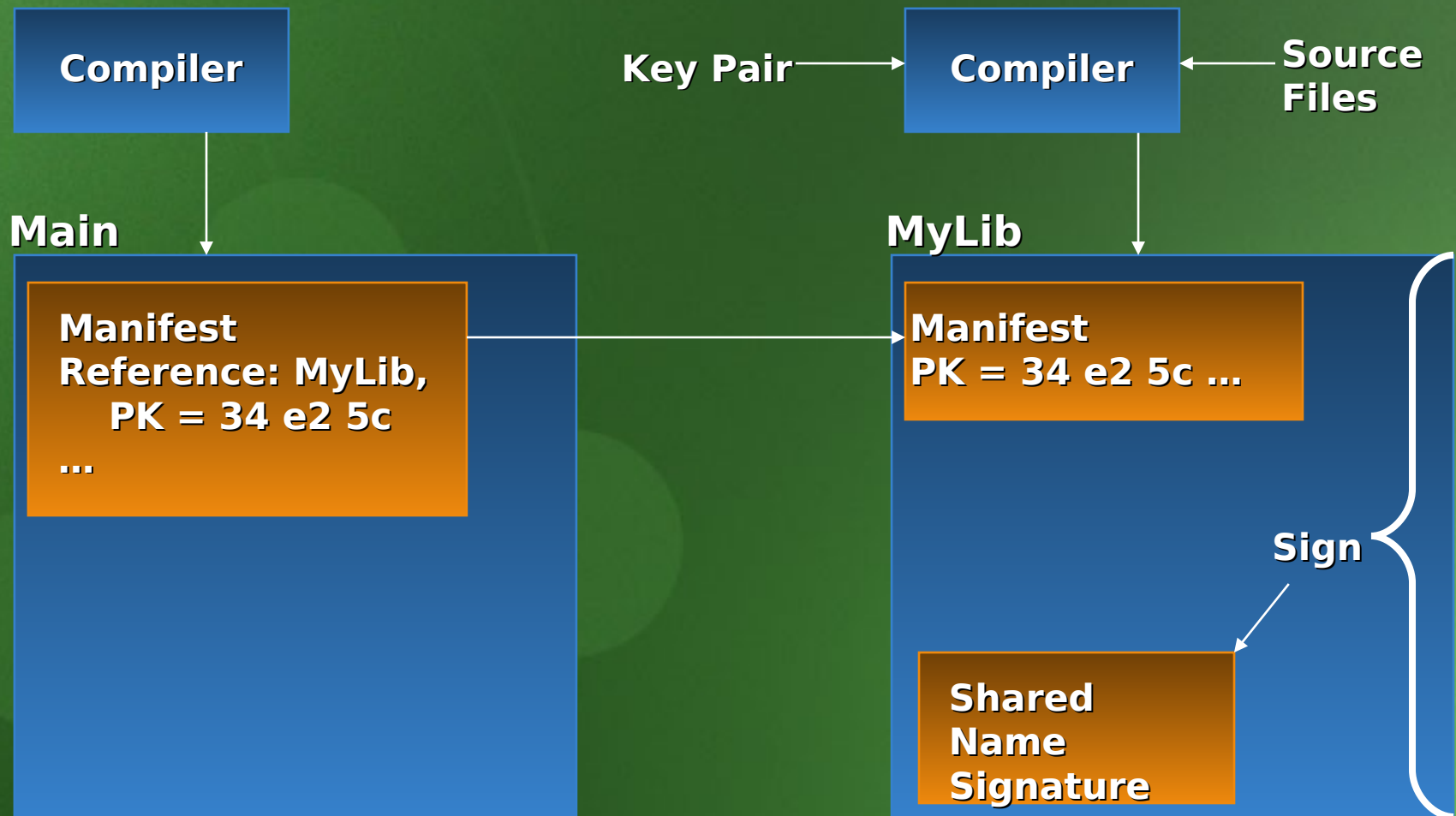
## Shared Assembly Names

### ■ Goals:

- Name uniqueness
  - Prevent spoofing
  - Provide identity on reference
- Basis for the solution: public key cryptography
  - Not related to Authenticode
  - Signing done by compiler or signing utility
  - “Test signed” during development

# Shared Assembly Names

## ■ Example: Main references Mylib



# Shared Assembly Names

- **SN.EXE** `sn -k orgKey.snk` **Process**
- **Assembly Linker (AL.EXE)**

```
csc /target:library ←  
  /out:Reverser\Reverser.dll ←  
  Reverser\Reverser.cs ←  
  /a.keyfile:orgKey.snk ←  
  /a.version:1.0.0.0
```

```
AL /install:Reverser\Reverser.dll
```

```
csc /target:library ←  
  /out:Stringer\Stringer.dll ←  
  Stringer\Stringer.cs
```

```
csc /reference:Stringer\Stringer.dll;Reverser\  
Reverser.dll ←
```



# Shared Assembly Names

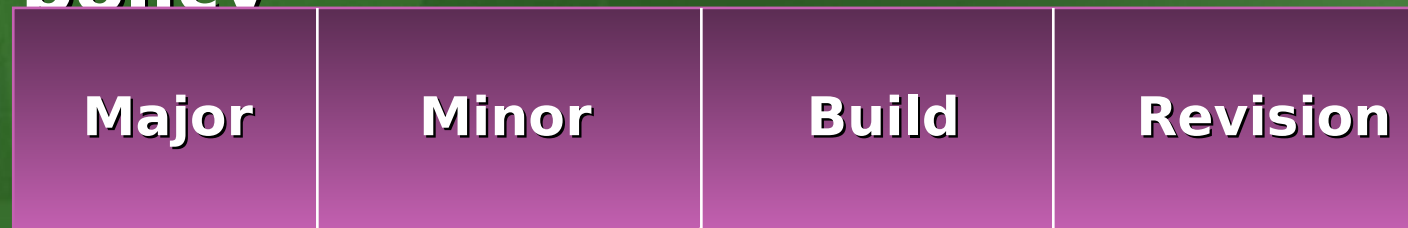
## At Runtime

- **MyLib's Name verified at install time (or load time)**
  - Public key stored in the assembly
  - Guarantees integrity
- **Main's reference to MyLib includes Public Key**
  - Recorded during development
  - Verifies MyLib came from the expected publisher

# Versioning

## The Version Number

- Assemblies have a physical 4-part version number
- Semantics applied to each part
- Plays key role in binding and version policy



Incompatible  
Change

Compatible Change: QFE (hotfix)

# **Version Policy**

## **Default Policy**

- **Manifest records the version of each dependency at build time**
- **Bind to Major.Minor, take the latest Build.Revision**
  - **Automatic QFE (hot fix) policy**
  - **QFE may come from app dir or global assembly store**
- **Applies only on references to shared assemblies**

# Demo: Default Policy

## Calculato

Calc 1.0

Math 1.0.0.0

<BindingMode>

<AppBindingMode

...

**Mode="Safe"/>**

</BindingMode>

Global  
Assembly  
Store

✓ Math 1.0.0.0

\$ Math 1.0.0.1 ✓

## Currency Converte

Convert

Math 1.0.0.1

# **Version Policy**

## **Custom Policy**

- **Policy can be altered using XML configuration files**
  - **Application-specific policy**
  - **Administrator policy**
- **Examples:**
  - **Bind to specific version**
  - **Turn off Automatic QFE policy**
  - **Safe Mode (run as-built)**



# Version Policy

## Stages in Resolution

### App Configuratio

```
<BindingPolicy  
Name = Math  
NewVersion = 2.0.0.0  
PK = 34 e2 3e ... />
```

### New Reference

```
Name = Math  
Version = 2.0.0.0  
PK = 34 e2 3e ... />
```

### Admin Configuration

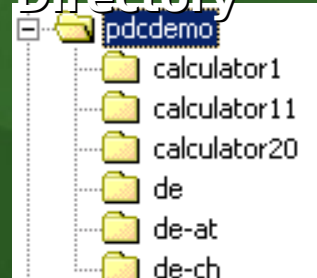
```
<BindingPolicy  
Name = Math  
NewVersion = 1.0.0.1  
/>
```

### App Domain

#### Calculator Manifest

```
Original Reference:  
Name = Math  
Version = 1.0.0.0  
PK = 34 e2 3e ...
```

### App Directory



Load  
Version  
2.0.0.0

Load  
Version  
1.0.0.1

### Global Assembly

Global A...	Version	Originator
calcr	1.0.0.0	32ab4ba45e0a69a1
calcr	1.0.0.1	32ab4ba45e0a69a1
calcr	2.0.0.0	32ab4ba45e0a69a1
gac	0.0.0.0	5036b1fd1f91d66c
gac	0.0.0.0	73eaad270abc3a03

# Demo: Custom Policy

Calcuato

Calc 1.0

Math 1.0.0.0

Currency  
Converte

Convert

Math 1.0.0.1

Global  
Assembly  
Store

✓ Math 1.0.0.0

\$ Math 1.0.0.1 ✓

Math 1.0.0.2 ✓

Math 1.0.0.3 ÷ 0

```
<BindingRedir
  Name="Math"
  Originator="..."
  Version="*"
  VersionNew="1.0.0
.2"
>
```

# **Deployment**

## **Packaging and Distribution**

- **Packaging**
  - **As built: dll, exe**
  - **Cab Files**
  - **Windows® Installer Package or other install format**
- **Distribution**
  - **XCOPY, FTP**
  - **Code download**
  - **Install program such as Windows**

# Deployment Scenario (1)

## Win Forms Application

- **Package: Windows Installer Package (.msi)**
- **Distribute: Windows Installer**
- **Deployment features:**
  - **Leverage Windows Installer and Windows 2000 App Mgmt**
    - **Group Policy Editor and the Active Directory™**
    - **Advertisement**
    - **Publishing, “Add/Remove Programs”**
    - **Repair**

# Deployment Scenario (2)

## Win Forms Control Download

- **Package: .cab or .dll**
- **Distribute: Code download**
- **Deployment Features:**
  - **No impact on client machine**
  - **File compression**
  - **Download isolated to the application**
  - **Download is incremental**



# Deployment Scenario

## (3) ASP+ Application

- **Package:** .aspx, .gif, .html, .dll
- **Distribute:** XCOPY or FTP to Web server
- **Deployment features:**
  - Zero impact: no registry
  - Remote application update
  - DLLs can be replaced on the fly!!
  - App can be moved or replicated by XCOPY

# Summary

## Key Takeaway Points

- **.NET Framework enables zero-impact installation and addresses “DLL Hell”**
- **Assemblies enable these features**
  - Self-describing, versionable deployment units
  - Shared names ensure name uniqueness and prevent spoofing
- **.NET Framework supports several packaging and distribution scenarios**
  - PE files, .cab, .msi
  - XCOPY, Windows Installer, code download

# **Related Sessions And References**

- **Other PDC sessions recommended:**
  - **“1-222 Security Considerations for Downloaded Controls.” Thurs at 10:15 in Room 304**
  - **“3-433 How Does Hello World Really Work?” Fri at 11:00 in Room 109**
- **Other SDK materials:**
  - **“Deployment” Whitepaper and Samples**

Where do **you** want to go today?

**Microsoft**